

On a Duality Between Recoverable Distributed Storage and Index Coding

Arya Mazumdar

Department of ECE

University of Minnesota– Twin Cities

Minneapolis, MN 55455

email: arya@umn.edu

Abstract—In this paper, we introduce a model of a single-failure locally recoverable distributed storage system. This model appears to give rise to a problem seemingly dual of the well-studied index coding problem. The relation between the dimensions of an optimal index code and optimal distributed storage code of our model has been established in this paper. We also show some extensions to vector codes.

I. INTRODUCTION

Recently, local repair property of error-correcting codes is the center of a lot of research activity. In a distributed storage system, a single server failure is the most common error-event, and in that case, the aim is to reconstruct the content of the server from as few other servers as possible (or by downloading minimal amount of data from other servers). The study of such *regenerative* storage systems was initiated in [9] and then followed up in several recent works. In [11], a particularly neat characterization of a local repair property is provided. It is assumed that, each symbol of an encoded message is stored at a different node in the network (since the symbol alphabet is unconstrained, a symbol could represent a packet or block of bits of arbitrary size). Accordingly, [11] investigates codes allowing any single symbol of any codeword to be recovered from at most a constant number of other symbols of the codeword, i.e., from a number of symbols that does not grow with the length of the code.

The work of [11] is then further generalized to several directions and a number of impossibility results regarding, as well as construction of, *locally repairable codes* were presented (see, for example, [5], [12], [17], [20], [22]), culminating in very recent construction of [21].

However, the topology of the network of distributed storage system is missing from the above definition of local repairability. Namely, all servers are treated equally irrespective of their physical positions, proximities, and connections. Here we take a step to include that into consideration. We study the case when the topology of the storage system is fixed and the network of storage is given by a graph. In our model, the servers are represented by the vertices of a graph, and two servers are connected by an edge if it is easier to establish up-or-down link between them, for reasons such as physical locations of the servers, architecture of the distributed system or homogeneity of softwares, etc. It turns out that, our model is

closely related to the following *index coding* problem on a side information graph. In this paper, we formalize this relation.

A. Index Coding

A very natural “source coding” problem on a network, called the *index coding*, was introduced in [3], and since then is a subject of extensive research. In the index coding problem a *side information* graph $G(V, E)$ is given. Each vertex $v \in V$ represents a receiver that is interested in knowing a uniform random variable $Y_v \in \mathbb{F}_q$. For any $v \in V$, define $N(v) = \{u \in V : (v, u) \in E\}$ to be the neighborhood of v . The receiver at v knows the values of the variables $Y_u, u \in N(v)$. How much information should a broadcaster transmit, such that every receiver knows the value of its desired random variable? Let us give the formal definition from [3], adapted for q -ary alphabet here.

Definition 1: An *index code* \mathcal{C} for \mathbb{F}_q^n with side information graph $G(V, E), V = \{1, 2, \dots, n\}$, is a set of codewords in \mathbb{F}_q^ℓ together with:

- 1) An encoding function f mapping inputs in \mathbb{F}_q^n to codewords, and
- 2) A set of deterministic decoding functions g_1, \dots, g_n such that $g_i(f(Y_1, \dots, Y_n), \{Y_j : j \in N(i)\}) = Y_i$ for every $i = 1, \dots, n$.

The encoding and decoding functions depend on G . The integer ℓ is called the length of \mathcal{C} , or $\text{len}(\mathcal{C})$. Given a graph G the minimum possible length of an index code is denoted by $\text{INDEX}_q(G)$.

In [3], a connection has been made with the length of an index code to a quantity called the minrank of the graph. Suppose, $A = (a_{ij})$ be an $n \times n$ matrix over \mathbb{F}_q . It is said that A *fits* $G(V, E)$ over \mathbb{F}_q if $a_{ii} \neq 0$ for all i and $a_{ij} = 0$ whenever $(i, j) \notin E$ and $i \neq j$.

Definition 2: The minrank of a graph $G(V, E)$ over \mathbb{F}_q is defined to be,

$$\text{minrank}_q(G) = \min\{\text{rank}_{\mathbb{F}_q}(A) : A \text{ fits } G\}. \quad (1)$$

It was shown in [3], that,

$$\text{INDEX}_q(G) \leq \text{minrank}_q(G), \quad (2)$$

and indeed, $\text{minrank}_q(G)$ is the minimum length of an index code on G when the encoding function, and the decoding

functions are all *linear*. The above inequality can be strict in many cases [1], [14].

In [1], the problem of index coding is further generalized. We only describe here what is important for our context. Just for this part, assume $q = 2$. To characterize the optimal size of an index code, [1] introduces the notion of a *confusion graph*. Two input strings, $\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n) \in \mathbb{F}_2^n$ are called *confusable* if there exists some $i \in \{1, \dots, n\}$, such that $x_i \neq y_i$, but $x_j = y_j$, for all $j \in N(i)$. In the confusion graph of G , total number of vertices are 2^n , and each vertex represents a different $\{0, 1\}$ -string of length n . There exists an edge between two vertices if and only if the corresponding two strings are confusable with respect to the graph G . The maximum size of an *independent set* of the confusion graph is denoted by $\gamma(G)$.

However, the confusion graph and $\gamma(G)$ in [1] were used as tools to characterize the *rate* of index coding; they were not used to model any immediate practical problem. In this paper, we show that, this notion of *confusable* strings fits perfectly to the situation of *local recovery* of a distributed storage system. Namely, $\gamma(G)$, in our problem becomes the largest possible size of a locally recoverable code for a system with topology given by G .

B. Organization

The paper is organized in the following way. In Section II, we introduce formally the model of a recoverable distributed storage system. The notion of an optimal recoverable distributed storage code given a graph and its relation to the optimal index code is also described here. In Section III, we provide an algorithmic proof of the main duality relation of the index code and distributed storage code. Our proof is based on a covering argument of the Hamming space, and rely on the fact that for any given subset of the Hamming space there exists a translation of the set, that has very small overlap with the original subset. We conclude with an extension of the duality theorem to vector codes and a remark on the optimal linearly recoverable distributed storage codes¹.

II. RECOVERABLE DISTRIBUTED STORAGE SYSTEMS

Consider the network of distributed storage, for example, one of Fig. 1. As mentioned in the introduction, the property of two servers connected by an edge is based on the ease of establishing a link between the servers. It is also possible (and sensible, perhaps) to model this as a directed graph (especially when uplink and downlink constructions have varying difficulties). In the following, we assume that the graph is directed, and an undirected graph is just a special case.

¹ After the first version of this paper appeared in arxiv, we were made aware of a parallel independent work [19] where for *vector linear codes* the duality between RDSS and index codes (see the discussion preceding Eq. (4)) is proved. The authors of [19] use that observation to give an upper bound on the optimal linear sum rate of the multiple unicast network coding problem. In this paper we have a different focus: we show a proof of (approximate) duality for general (nonlinear) codes.

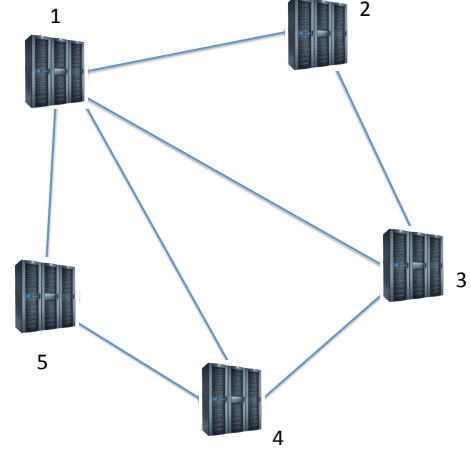


Fig. 1. Example of a distributed storage graph

If the data of any one server is lost, we want to recover it from the *nearby* servers, i.e., the ones with which it is easy to establish a link. This notion is formalized below.

Suppose, the directed graph $G(V, E)$ represents the network of storage. Each element of V represents a server, and in the case of a server failure (say, $v \in V$ is the failed server) one must be able to reconstruct its content from its neighborhood $N(v)$.

Given, this constraint what is the maximum amount of information one can store in the system? Without loss of generality, assume $V = \{1, 2, \dots, n\}$ and the variables X_1, X_2, \dots, X_n respectively denote the content of the vertices, where, $X_i \in \mathbb{F}_q, i = 1, \dots, n$.

Definition 3: A *recoverable distributed storage system (RDSS)* code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with storage recovery graph $G(V, E), V = \{1, 2, \dots, n\}$, is a set of vectors in \mathbb{F}_q^n together with:

- A set of deterministic recovery functions, $f_i : \mathbb{F}_q^{N(i)} \rightarrow \mathbb{F}_q$ for $i = 1, \dots, n$ such that for any codeword $(X_1, X_2, \dots, X_n) \in \mathbb{F}_q^n$,

$$X_i = f_i(\{X_j : j \in N(i)\}), \quad i = 1, \dots, n. \quad (3)$$

Again, the decoding functions depend on G . The log-size of the code, $\log_q |\mathcal{C}|$, is called the dimension of \mathcal{C} , or $\dim(\mathcal{C})$. Given a graph G the maximum possible dimension of an RDSS code is denoted by $\text{RDSS}_q(G)$.

For example, consider the graph of Fig. 1 again. Here, $V = \{1, 2, 3, 4, 5\}$. The recovery sets of each vertex (or storage nodes) are given by:

$$\begin{aligned} N(1) &= \{2, 3, 4, 5\}, N(2) = \{1, 3\}, N(3) = \{1, 2, 4\}, \\ N(4) &= \{1, 3, 5\}, N(5) = \{1, 4\}. \end{aligned}$$

Suppose, the contents of the nodes $1, 2, \dots, 5$ are X_1, X_2, \dots, X_5 respectively, where, $X_i \in \mathbb{F}_q, i = 1, \dots, 5$. Moreover, $X_1 = f_1(X_2, X_3, X_4, X_5), X_2 = f_2(X_1, X_3), X_3 = f_3(X_1, X_2, X_4), X_4 = f_4(X_1, X_3, X_5), X_5 = f_5(X_1, X_4)$.

Assume, the functions $f_i, i = 1, \dots, 5$, in this example are linear. That is, for $\alpha_{ij} \in \mathbb{F}_q, 1 \leq i, j \leq 5$,

$$\begin{aligned} X_1 &= \alpha_{12}X_2 + \alpha_{13}X_3 + \alpha_{14}X_4 + \alpha_{15}X_5 \\ X_2 &= \alpha_{21}X_1 + \alpha_{23}X_3 \\ X_3 &= \alpha_{31}X_1 + \alpha_{32}X_2 + \alpha_{34}X_4 \\ X_4 &= \alpha_{41}X_1 + \alpha_{43}X_3 + \alpha_{45}X_5 \\ X_5 &= \alpha_{51}X_1 + \alpha_{54}X_4. \end{aligned}$$

This implies, (X_1, X_2, \dots, X_5) must belong to the null-space (over \mathbb{F}_q) of

$$D \equiv \begin{pmatrix} 1 & -\alpha_{12} & -\alpha_{13} & -\alpha_{14} & -\alpha_{15} \\ -\alpha_{21} & 1 & -\alpha_{23} & 0 & 0 \\ -\alpha_{31} & -\alpha_{32} & 1 & -\alpha_{34} & 0 \\ -\alpha_{41} & 0 & -\alpha_{43} & 1 & -\alpha_{45} \\ -\alpha_{51} & 0 & 0 & -\alpha_{54} & 1 \end{pmatrix}.$$

The dimension of the null-space of D is n minus the rank of D . Hence, it is evident that the dimension of the RDSS code is $n - \text{minrank}_q(G)$. Also, the null-space of a linear index code for G is a linear RDSS code for the same graph G (see, Eq. (2)). From the above discussion, we have,

$$\text{RDSS}_q(G) \geq n - \text{minrank}_q(G), \quad (4)$$

and, $n - \text{minrank}_q(G)$ is the maximum possible dimension of an RDSS code when the recovery functions are all linear. At this point, it is tempting to make the assertion $\text{RDSS}_q(G) = n - \text{INDEX}_q(G)$, however, that would be wrong. This is shown in the following example.

This example is present in [1], and the distributed storage graph, a *pentagon*, is shown in Fig. 2. For this graph, a maximum-sized binary RDSS code consists of the codewords $\{00000, 01100, 00011, 11011, 11101\}$. The recovery functions are given by,

$$\begin{aligned} X_1 &= X_2 \wedge X_5, X_2 = X_1 \vee X_3, X_3 = X_2 \wedge \bar{X}_4, \\ X_4 &= \bar{X}_3 \wedge X_5, X_5 = X_1 \vee X_4. \end{aligned}$$

If all the recovery functions are linear, we could not have an RDSS code with so many codewords. Here $\text{RDSS}_2(G) = \log_2 5$. On the other hand, the minimum length of an index code for this graph is 3, i.e., $\text{INDEX}_2(G) = 3$, and this is achieved by the following linear mappings. The broadcaster transmit $Y_1 = X_2 + X_3, Y_2 = X_4 + X_5$ and $Y_3 = X_1 + X_2 + X_3 + X_4 + X_5$. The decoding functions are, $X_1 = Y_1 + Y_2 + Y_3, X_2 = Y_1 + X_3, X_3 = Y_1 + X_2, X_4 = Y_2 + X_5, X_5 = Y_2 + X_4$.

Although in general $\text{RDSS}_q(G) \neq n - \text{INDEX}_q(G)$, these two quantities are not too far from each other. In particular, for large enough alphabet, the left and right hand sides can be arbitrarily close. This is reflected in Thm. 1 below.

It is to be noted that, we refrain from using ceiling and floor functions for clarity in this paper. In many cases, it is clear that the number of interest is not an integer and should be rounded off to the nearest larger or smaller integer. The main results do not change for this.

A. Implication of the results of [1]

The result of [1] can be cast in our context in the following way.

Theorem 1: Given a graph $G(V, E)$, we must have,

$$\begin{aligned} n - \text{RDSS}_q(G) &\leq \text{INDEX}_q(G) \leq n - \text{RDSS}_q(G) \\ &+ \log_q \left(\min\{n \ln q, 1 + \text{RDSS}_q(G) \ln q\} \right). \end{aligned} \quad (5)$$

This result is purely graph-theoretic, the way it was presented in [1]. In particular, the size of maximum independent set of the confusion graph, $\gamma(G)$ was identified as the size of the RDSS code, and its relation to the *chromatic number* of the confusion graph, which represents the size of the index code was found. Namely the proof was dependent on the following two crucial steps.

- 1) The *chromatic number* of the graph can only be so much away from the *fractional chromatic number* (see, [1] for detailed definition).
- 2) The confusion graph is *vertex transitive*. This implies that the maximum size of an independent set is equal to the number of vertices divided by the fractional chromatic number.

A proof of the first fact above can be found in [13]. In what follows, we give a simple *coding theoretic* proof of this main theorem, without using the notion of the confusion graph or its vertex transitivity, for completeness.

III. THE PROOF OF THE DUALITY

We prove Theorem 1 with the help of following two lemmas. The first of them is immediate.

Lemma 2: If there exists an index code \mathcal{C} of length ℓ for a side information graph G on n vertices, then there exists an RDSS code of dimension $n - \ell$ for the distributed storage graph G .

Proof: Suppose, the encoding and decoding functions of the index code \mathcal{C} are $f : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^\ell$ and $g_i : \mathbb{F}_q^{\ell + N(i)} \rightarrow \mathbb{F}_q, i = 1, \dots, n$. There must exist some $\mathbf{x} \in \mathbb{F}_q^\ell$ such that $|\{\mathbf{y} \in \mathbb{F}_q^n : f(\mathbf{y}) = \mathbf{x}\}| \geq q^{n-\ell}$. Let, $\mathcal{D}_x \equiv \{\mathbf{y} \in \mathbb{F}_q^n : f(\mathbf{y}) = \mathbf{x}\}$ be the distributed storage code with recovery functions,

$$f_i(\{X_j, j \in N(i)\}) \equiv g_i(\mathbf{x}, \{X_j, j \in N(i)\}).$$

The second lemma is the more interesting one.

Lemma 3: If there exists an RDSS code \mathcal{C} of dimension k for a distributed storage graph G on n vertices, then there exists an index code of length $n - k + \log_q \min\{n \ln q, 1 + k \ln q\}$ for the side information graph G .

To prove this result, we need the help of a number of other lemmas. First of all notice that, translation of any RDSS code is an RDSS code.

Lemma 4: Suppose, $\mathcal{C} \subseteq \mathbb{F}_q^n$ is an RDSS code. Then any known translation of \mathcal{C} is also an RDSS code of same dimension. That is, for any $\mathbf{a} \in \mathbb{F}_q^n, \mathcal{C} + \mathbf{a} \equiv \{\mathbf{y} + \mathbf{a} : \mathbf{y} \in \mathcal{C}\}$ is an RDSS code of dimension $\log_q |\mathcal{C}|$.

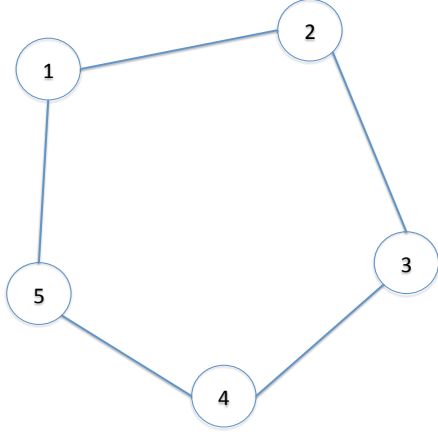


Fig. 2. A distributed storage graph (the pentagon) that shows $\text{RDSS}(\mathcal{G}) \neq n - \text{INDEX}(\mathcal{G})$.

Proof: Let, $(X_1, \dots, X_n) \in \mathcal{C}$. Also assume, $\mathbf{a} = (a_1, \dots, a_n)$, and $X'_i = X_i + a_i$. We know that, there exist recovery functions such that, $X_i = f_i(\{X_j : j \in N(i)\})$.

Now, $X'_i = X_i + a_i = f_i(\{X_j : j \in N(i)\}) + a_i \equiv f'_i(\{X'_j : j \in N(i)\})$. ■

In particular, Lemma 3 crucially use the existence of a covering of the entire \mathbb{F}_q^n , by translations of an RDSS code.

Proof of Lemma 3: We will show that there exists, $\mathcal{C}_1, \dots, \mathcal{C}_m$, $\mathcal{C}_i \in \mathbb{F}_q^n$, $i = 1, \dots, m$, all of which are RDSS codes of dimension k such that

$$\bigcup_{i=1}^m \mathcal{C}_i = \mathbb{F}_q^n, \quad (6)$$

where $m = q^{n-k} \min\{n \ln q, 1 + k \ln q\}$. Assume, the above is true. Then, any $\mathbf{y} \in \mathbb{F}_q^n$ must belong to at least one of the \mathcal{C}_i s. Suppose, $\mathbf{y} \equiv (Y_1, \dots, Y_n) \in \mathbb{F}_q^n$ and $\mathbf{y} \in \mathcal{C}_i$. Then, the encoding function of the desired index code \mathcal{D} is simply given by, $f(\mathbf{y}) = i$. If the recovery functions of \mathcal{C}_i are f_j^i , $j = 1, \dots, n$, then, the decoding functions of \mathcal{D} are given by:

$$g_j(i, \{Y_l : l \in N(j)\}) = f_j^i(\{Y_l : l \in N(j)\}).$$

Clearly the length of the index code is $\log_q m = n - k + \log_q(\min\{n \ln q, 1 + k \ln q\})$.

It remains to show the existence of RDSS codes $\mathcal{C}_1, \dots, \mathcal{C}_m$ of dimension k each with property (6). We will show that, there exists m vectors \mathbf{x}_j , $j = 1, \dots, m$ such that

$$\mathcal{C}_i = \mathcal{C} + \mathbf{x}_i \equiv \{\mathbf{y} + \mathbf{x}_i : \mathbf{y} \in \mathcal{C}\}. \quad (7)$$

From Lemma 4, \mathcal{C}_i , $i = 1, \dots, m$ are all RDSS codes of dimension k . Suppose, \mathbf{x}_i , $i = 1, \dots, m$ are randomly and independently chosen from \mathbb{F}_q^n . Now,

$$\Pr(\bigcup_{i=1}^{m'} \mathcal{C}_i \neq \mathbb{F}_q^n) \leq q^n (1 - |\mathcal{C}|/q^n)^{m'} < q^n e^{-m'|\mathcal{C}|/q^n} \leq 1,$$

when we set $m' = q^{n-k} n \ln q \leq m$ in the above expression (see [2, Prop. 3.12]).

If, instead we set $m' = q^{n-k} k \ln q$ then, $\Pr(\bigcup_{i=1}^{m'} \mathcal{C}_i \neq \mathbb{F}_q^n) \leq q^{n-k}$, which is also the expected number of points, that do not belong to any of the m' translations. To cover all these remaining points we need at most q^{n-k} other transmission. Hence, there must exists a covering such that $q^{n-k} k \ln q + q^{n-k} = q^{n-k} (k \ln q + 1) \leq m$ translations suffice. ■

The proof of Lemma 3 can also be given via a greedy algorithm. In the greedy algorithm about $\log m$ vectors are recursively chosen instead of m random vectors. We provide the construction/proof next.

A. A greedy algorithm for the proof of Lemma 3

Note that, to proof Lemma 3 we need to show the existence of a covering of the entire \mathbb{F}_q^n , by translations of an RDSS code. What we show here is that the translations themselves form a linear subspace. The greedy covering argument that we employ below was used to show the existence of good linear covering codes in [8] (see, also, [7], [10]), and was reintroduced in [15] to show the existence of *balancing sets*.

Lemma 5 (Bassalygo-Elias): Suppose, $\mathcal{C}, \mathcal{B} \subseteq \mathbb{F}_q^n$. Then,

$$\sum_{\mathbf{x} \in \mathbb{F}_q^n} |(\mathcal{C} + \mathbf{x}) \cap \mathcal{B}| = |\mathcal{C}||\mathcal{B}|. \quad (8)$$

Proof:

$$\begin{aligned} \sum_{\mathbf{x} \in \mathbb{F}_q^n} |(\mathcal{C} + \mathbf{x}) \cap \mathcal{B}| &= |\{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbb{F}_q^n, \mathbf{y} \in \mathcal{B}, \mathbf{y} \in \mathcal{C} + \mathbf{x}\}| \\ &= |\{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbb{F}_q^n, \mathbf{y} \in \mathcal{B}, \mathbf{x} \in \mathbf{y} - \mathcal{C}\}| \\ &= |\{(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in \mathcal{B}, \mathbf{x} \in \mathbf{y} - \mathcal{C}\}| \\ &= |\mathcal{B}||\mathbf{y} - \mathcal{C}| = |\mathcal{C}||\mathcal{B}|, \end{aligned}$$

where $\mathbf{y} - \mathcal{C} \equiv \{\mathbf{y} - \mathbf{a} : \mathbf{a} \in \mathcal{C}\}$. ■

For any set $\mathcal{F} \subseteq \mathbb{F}_q^n$, define

$$Q(\mathcal{F}) \equiv 1 - \frac{|\mathcal{F}|}{q^n}. \quad (9)$$

In words, $Q(\mathcal{F})$ denote the proportion of \mathbb{F}_q^n that is not covered by \mathcal{F} . The following property is a result of Lemma 5.

Lemma 6: For every subset $\mathcal{F} \subseteq \mathbb{F}_q^n$,

$$q^{-n} \sum_{\mathbf{x} \in \mathbb{F}_q^n} Q(\mathcal{F} \cup (\mathcal{F} + \mathbf{x})) = Q(\mathcal{F})^2. \quad (10)$$

Proof: We have,

$$|\mathcal{F} \cup (\mathcal{F} + \mathbf{x})| = 2|\mathcal{F}| - |\mathcal{F} \cap (\mathcal{F} + \mathbf{x})|.$$

Therefore,

$$Q(\mathcal{F} \cup (\mathcal{F} + \mathbf{x})) = 1 - 2|\mathcal{F}|q^{-n} + |\mathcal{F} \cap (\mathcal{F} + \mathbf{x})|q^{-n},$$

and hence,

$$\begin{aligned} q^{-n} \sum_{\mathbf{x} \in \mathbb{F}_q^n} Q(\mathcal{F} \cup (\mathcal{F} + \mathbf{x})) &= 1 - 2|\mathcal{F}|q^{-n} \\ &\quad + q^{-2n} \sum_{\mathbf{x} \in \mathbb{F}_q^n} |\mathcal{F} \cap (\mathcal{F} + \mathbf{x})| \\ &= 1 - 2|\mathcal{F}|q^{-n} + q^{-2n} |\mathcal{F}|^2 \end{aligned}$$

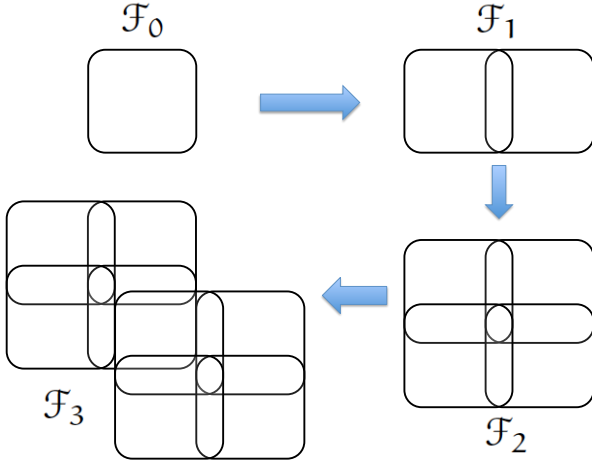


Fig. 3. The recursive construction of the sets $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ of Lemma 7.

$$= (1 - |\mathcal{F}|q^{-n})^2,$$

where in the second line we have used Lemma 5. ■

The implication of the above lemma is the following result.

Lemma 7: For every subset $\mathcal{F} \subseteq \mathbb{F}_q^n$, there exists $m = q^n |\mathcal{F}|^{-1} \min\{n \ln q, 1 + \ln |\mathcal{F}|\}$ vectors $\mathbf{x}_0 = 0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{m-1} \in \mathbb{F}_q^n$, such that

$$\cup_{i=0}^{m-1} (\mathcal{F} + \mathbf{x}_i) = \mathbb{F}_q^n.$$

Proof: From Lemma 6, for every subset $\mathcal{F} \subseteq \mathbb{F}_q^n$, there exists $\mathbf{x} \in \mathbb{F}_q^n$ such that

$$Q(\mathcal{F} \cup (\mathcal{F} + \mathbf{x})) \leq Q(\mathcal{F})^2.$$

For the set $\mathcal{F} \equiv \mathcal{F}_0$, recursively define, for $i = 1, 2, \dots$

$$\mathcal{F}_i = \mathcal{F}_{i-1} \cup (\mathcal{F}_{i-1} + \mathbf{z}_{i-1}),$$

where $\mathbf{z}_i \in \mathbb{F}_q^n$ is such that,

$$Q(\mathcal{F}_i \cup (\mathcal{F}_i + \mathbf{z}_i)) \leq Q(\mathcal{F}_i)^2, \quad i = 0, 1, \dots$$

Clearly,

$$Q(\mathcal{F}_t) \leq Q(\mathcal{F}_0)^{2^t} = (1 - q^{-n} |\mathcal{F}|)^{2^t} \leq e^{-q^{-n} |\mathcal{F}| 2^t}.$$

At this point we can just use the argument at the end of proof of Lemma 3, with 2^t plating the role of m' .

On the other hand \mathcal{F}_t contains \mathcal{F}_0 and its $2^t - 1$ translations (see, Figure 3 for an illustration). Hence, there exists $m = \min \left\{ \frac{q^n n \ln q}{|\mathcal{F}|}, \frac{q^n (1 + \ln |\mathcal{F}|)}{|\mathcal{F}|} \right\}$ vectors $\mathbf{x}_0 = 0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{m-1} \in \mathbb{F}_q^n$, such that

$$\cup_{i=0}^{m-1} (\mathcal{F} + \mathbf{x}_i) = \mathbb{F}_q^n.$$

To complete the proof of Lemma 3, as before, we just show the existence of RDSS codes $\mathcal{C}_0 \equiv \mathcal{C}, \mathcal{C}_1, \dots, \mathcal{C}_{m-1}$

of dimension k each with property (6). This is achieved by choosing $m - 1$ vectors $\mathbf{x}_j, j = 1, \dots, m - 1$ such that

$$\mathcal{C}_i = \mathcal{C} + \mathbf{x}_i \equiv \{\mathbf{y} + \mathbf{x}_i : \mathbf{y} \in \mathcal{C}\}. \quad (11)$$

From Lemma 4, $\mathcal{C}_i, i = 1, \dots, m - 1$ are all RDSS codes of dimension k . Moreover, from Lemma 7, we already know the existence of $\mathbf{x}_j, j = 1, \dots, m - 1$ such that property (6) is satisfied. However, from Lemma 7 it is also clear that these m vectors form a linear subspace and can be generated by only $\log_q m$ vectors.

Corollary 8: For every subset $\mathcal{F} \subseteq \mathbb{F}_q^n$, there exists a linear subspace $\mathcal{D} \in \mathbb{F}_q^n$ such that $|\mathcal{D}| = q^n |\mathcal{F}|^{-1} n \ln q$ and

$$\cup_{\mathbf{x} \in \mathcal{D}} (\mathcal{F} + \mathbf{x}) = \mathbb{F}_q^n.$$

The above result is helpful in the decoding process of the index code. If \mathcal{C} is an RDSS code and \mathcal{D} is the linear subspace such that $\cup_{\mathbf{x} \in \mathcal{D}} (\mathcal{C} + \mathbf{x}) = \mathbb{F}_q^n$, then the decoding of the obtained index code can be performed from $\mathbf{x} \in \mathbb{F}_q^{\log_q |\mathcal{D}|}$ by first multiplying \mathbf{x} with the generator matrix of \mathcal{D} and then shifting \mathcal{C} by it. Hence, if there is a polynomial time decoding algorithm for \mathcal{C} then there will be one for the index code. It would not be so for the case of random-choice, where we must maintain a look-up table of size exponential in n .

Remark 1: It is a perhaps not so surprising that the method of [1], that is the random choice, (or, in fact the method of [13]) gives the exact same result as the greedy algorithm method.

IV. EXTENSION TO VECTOR CODES AND THE CAPACITY OF LINEAR CODES

Literatures of distributed storage often considers *vector linear codes* and the same is true for [19]. However in the context of general nonlinear codes, vector codes do not bring any further technical novelty and can just be thought of as codes over a larger alphabet.

For vector index codes, as earlier, a *side information* graph $G(V, E)$ is given. Each vertex $v \in V$ represents a receiver that is interested in knowing a uniform random vector $Y_v \in \mathbb{F}_q^p$. The receiver at v knows the values of the variables $Y_u, u \in N(v)$. A vector *index code* \mathcal{C} for \mathbb{F}_q^{np} with side information graph $G(V, E), V = \{1, 2, \dots, n\}$, is a set of codewords in $\mathbb{F}_q^{\ell p}$ (ℓ is the length of the code) together with:

- 1) An encoding function f mapping inputs in \mathbb{F}_q^{np} to codewords, and
- 2) A set of deterministic decoding functions g_1, \dots, g_n such that $g_i(f(Y_1, \dots, Y_n), \{Y_j : j \in N(i)\}) = Y_i$ for every $i = 1, \dots, n$.

Given a graph G the minimum possible value of ℓ is denoted by $\text{INDEX}_q^p(G)$ (also called the *broadcast capacity*). When the function f, g_i are linear, for all $1 \leq i \leq n$, in all of their arguments in \mathbb{F}_q , then the code is called *vector linear*.

Similar generalization is possible for the definition of RDSS codes. A vector RDSS code $\mathcal{C} \subseteq (\mathbb{F}_q^p)^n$ with storage recovery graph $G(V, E), V = \{1, 2, \dots, n\}$, is a set of vectors in \mathbb{F}_q^{np} together with: A set of deterministic recovery functions, $f_i :$

$\mathbb{F}_q^{N(i)p} \rightarrow \mathbb{F}_q^p$ for $i = 1, \dots, n$ such that for any codeword (X_1, X_2, \dots, X_n) , $X_i \in \mathbb{F}_q^p$,

$$X_i = f_i(\{X_j : j \in N(i)\}), \quad i = 1, \dots, n. \quad (12)$$

The normalized log-size of the code, $\frac{1}{p} \log_q |\mathcal{C}|$, is called the dimension of \mathcal{C} . Given a graph G the maximum possible dimension of a vector RDSS code is denoted by $\text{RDSS}_q^p(G)$. When the decoding functions f_i , $1 \leq i \leq n$ are linear in all their arguments (in \mathbb{F}_q), the code is called *vector linear*.

General (nonlinear) vector index or RDSS codes can also be thought as scalar codes over the alphabet of size q^p . Hence,

$$\begin{aligned} n - \text{RDSS}_q^p(G) &\leq \text{INDEX}_q^p(G) \\ &\leq n - \text{RDSS}_q^p(G) + \frac{\log_q(pn \ln q)}{p}. \end{aligned}$$

As a consequence, even for a constant q , if $p = \Omega(\log n)$, we have $\text{INDEX}_q^p(G)$ and $n - \text{RDSS}_q^p(G)$ differ at most by 1 for any graph G – and for larger p , this difference goes to zero.

Although, general vector codes do not lead to a different analysis, we next show that vector linear codes can achieve a dimension sufficiently close to $\text{RDSS}_q^p(G)$ for any graph $G(V, E)$. This should be put into contrast with results, such as [4, Thm. 1.2], which show that a rather large gap must exist between vector linear and nonlinear index coding (or network coding) rates.

Proposition 9: There exists a polynomial time (in n) constructible vector linear RDSS code with dimension at least

$$\frac{\text{RDSS}_q^p(G)}{\beta \log n \cdot \log \log n}$$

for a large enough integer p and a constant $\beta < 5$.

Proof: In [19], it was shown that the linear algebraic dual of a vector linear index code is a vector linear RDSS code (see, Section II of this paper for scalar codes). This implies that, for a vector linear index code of length ℓ , the dual code is a vector linear RDSS code of dimension $n - \ell$. In [6], a vector linear index code of length ℓ was constructed in polynomial time, such that $n - \ell \geq \frac{n - \text{INDEX}_q^p(G)}{\alpha \log n \cdot \log \log n}$, (this result of [6] was also used in [19]), α is a constant (see, [18], the building-block of [6], for the value of the constant). The dual code of this code must be a vector RDSS code of dimension $k = n - \ell$. From the above discussion, it is evident that,

$$n - \text{INDEX}_q^p(G) \geq \text{RDSS}_q^p(G) - \frac{\log_q(pn \ln q)}{p}.$$

Hence,

$$k \geq \frac{\text{RDSS}_q^p(G) - \frac{\log_q(pn \ln q)}{p}}{\alpha \log n \cdot \log \log n}.$$

Hence if p is large enough, then the statement of the theorem is proved. ■

Remark 2: How large does p needs to be for the above proposition to hold? It is clear that $p = \Omega(\log n)$ is enough to diminish the additive error term of $\frac{\log_q(pn \ln q)}{p}$. However, for the algorithm of [6] to work, p needs to be as large as the denominator of a linear programming solution (see, [16]) that

is used crucially in [6]. Hence p , depending on the number of cycles in the graph, may required to be exponential in n .

Acknowledgement: The author thanks A. Agarwal, A. G. Dimakis and K. Shanmugam for useful references. This work was supported in part by a grant from University of Minnesota.

REFERENCES

- [1] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim. Broadcasting with side information. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 823–832. IEEE, 2008.
- [2] L. Babai. Automorphism groups, isomorphism, and reconstruction, chapter 27 of handbook of combinatorics. *North-Holland-Elsevier*, pages 1447–1540, 1995.
- [3] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol. Index coding with side information. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 197–206. IEEE, 2006.
- [4] A. Blasiak, R. Kleinberg, and E. Lubetzky. Lexicographic products and the power of non-linear network coding. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 609–618. IEEE, 2011.
- [5] V. Cadambe and A. Mazumdar. An upper bound on the size of locally recoverable codes. In *Proc. IEEE Int. Symp. Network Coding*, June 2013.
- [6] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg. On the complementary index coding problem. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 244–248. IEEE, 2011.
- [7] G. Cohen. A nonconstructive upper bound on covering radius. *Information Theory, IEEE Transactions on*, 29(3):352–353, 1983.
- [8] P. Delsarte and P. Piret. Do most binary linear codes achieve the gobblick bound on the covering radius?(corresp.). *Information Theory, IEEE Transactions on*, 32(6):826–828, 1986.
- [9] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inform. Theory*, 56(9):4539–4551, Sep. 2010.
- [10] T. J. Gobblick. *Coding for a discrete information source with a distortion measure*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [11] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin. On the locality of codeword symbols. *IEEE Trans. Inform. Theory*, 58(11):6925–6934, Nov. 2012.
- [12] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar. Codes with local regeneration. *arXiv preprint arXiv:1211.1932*, 2012.
- [13] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete mathematics*, 13(4):383–390, 1975.
- [14] E. Lubetzky and U. Stav. Nonlinear index coding outperforming the linear optimum. *Information Theory, IEEE Transactions on*, 55(8):3544–3551, 2009.
- [15] A. Mazumdar, R. M. Roth, and P. O. Vontobel. On linear balancing sets. *Advances in Mathematics of Communications (AMC)*, 4(3):345–361, 2010.
- [16] Z. Nutov and R. Yuster. Packing directed cycles efficiently. In *Mathematical Foundations of Computer Science 2004*, pages 310–321. Springer, 2004.
- [17] D. S. Papailiopoulos and A. G. Dimakis. Locally repairable codes. In *Proc. Int. Symp. Inform. Theory*, pages 2771–2775, Cambridge, MA, July 2012.
- [18] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.
- [19] K. Shanmugam and A. G. Dimakis. Connections between index coding, locally repairable codes and the multiple unicast problem. *personal communication*, 2014.
- [20] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath. Optimal locally repairable codes via rank-metric codes. *preprint, arXiv:1301.6331*, 2013.
- [21] I. Tamo and A. Barg. A family of optimal locally recoverable codes. *arXiv preprint arXiv:1311.3284*, 2013.
- [22] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis. Optimal locally repairable codes and connections to matroid theory. *preprint, arXiv:1301.7693*, 2013.